

“Cloudera’s instructor was excellent, offering clear and concise training that was easy to understand. His wide-ranging peripheral knowledge helped apply the course materials to real-world situations. I look forward to attending another course.”

Comscore

Cloudera Training for Apache HBase

Overview

Cloudera University’s three-day training course for Apache HBase enables participants to store and access massive quantities of multi-structured data and perform hundreds of thousands of operations per second.

Learn a modern toolset

Apache HBase is a distributed, scalable, NoSQL database built on Apache Hadoop. HBase can store data in massive tables consisting of billions of rows and millions of columns, serve data to many users and applications in real time, and provide fast, random read/write access to users and applications.

Get hands-on experience

Through instructor-led discussion and interactive, hands-on exercises, participants will navigate the Hadoop ecosystem, learning topics such as:

- The use cases and usage occasions for HBase, Hadoop, and RDBMS
- Using the HBase shell to directly manipulate HBase tables
- Designing optimal HBase schemas for efficient data storage and recovery
- How to connect to HBase using the Java API to insert and retrieve data in real time
- Best practices for identifying and resolving performance bottlenecks

What to expect

This course is appropriate for developers and administrators who intend to use HBase. Prior experience with databases and data modeling is helpful, but not required. Knowledge of Java is assumed. Prior knowledge of Hadoop is not required, but Cloudera Developer Training for Apache Hadoop provides an excellent foundation for this course.

Course Outline: Cloudera Training for Apache HBase

Introduction

Introduction to Hadoop and HBase

- Introducing Hadoop
- Core Hadoop Components
- What Is HBase?
- Why Use HBase?
- Strengths of HBase
- HBase in Production
- Weaknesses of HBase

HBase Tables

- HBase Concepts
- HBase Table Fundamentals
- Thinking About Table Design

HBase Shell

- Creating Tables with the HBase Shell
- Working with Tables
- Working with Table Data

HBase Architecture Fundamentals

- HBase Regions
- HBase Cluster Architecture
- HBase and HDFS Data Locality

HBase Schema Design

- General Design Considerations
- Application-Centric Design
- Designing HBase Row Keys
- Other HBase Table Features

Basic Data Access with the HBase API

- Options to Access HBase Data
- Creating and Deleting HBase Tables
- Retrieving Data with Get
- Retrieving Data with Scan
- Inserting and Updating Data
- Deleting Data

More Advanced HBase API Features

- Filtering Scans
- Best Practices
- HBase Coprocessors

HBase Write Path

- HBase Write Path
- Compaction
- Splits

HBase Read Path

- How HBase Reads Data
- Block Caches for Reading

HBase Performance Tuning

- Column Family Considerations
- Schema Design Considerations
- Configuring for Caching
- Memory Considerations
- Dealing with Time Series and Sequential Data
- Pre-Splitting Regions

HBase Administration and Cluster Management

- HBase Daemons
- ZooKeeper Considerations
- HBase High Availability
- Using the HBase Balancer
- Fixing Tables with hbck
- HBase Security

HBase Replication and Backup

- HBase Replication
- HBase Backup
- MapReduce and HBase Clusters

Using Hive and Impala with HBase

- How to Use Hive and Impala to Access HBase

Conclusion

Appendix A: Accessing Data with Python and Thrift

- Thrift Usage
- Working with Tables
- Getting and Putting Data
- Scanning Data
- Deleting Data
- Counters
- Filters

Appendix B: OpenTSDB